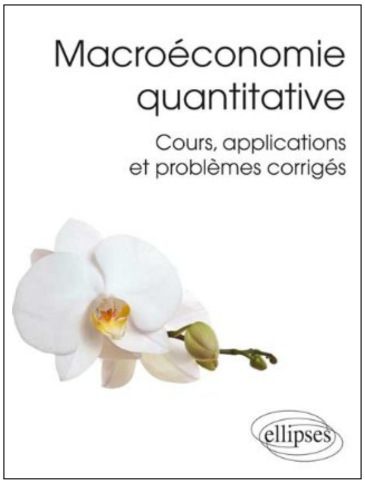


Application Chapitre 4 (corrigé)

Modèle VAR estimé sur la zone euro

Dans ce document, je propose une correction de l'application du Chapitre 4 de mon manuel de Macroéconomie quantitative (p.55). Les fonctions et les données utiles sont disponibles sur mon site[a]. Un ou plusieurs scripts Octave/Matlab accompagnent également ce document, dont une partie des éléments qui sont repris et commentés ici, apparaissent sous une police à `chasse fixe`.

a. www.christophecahn.fr/macroquant

Étude des séries pour la zone euro

Le fichier `./data/ZEdata.txt` contient des séries trimestrielles du produit intérieur brut (PIB) en volume, de l'inflation (déflateur du PIB) et du taux d'intérêt nominal à court terme de la zone euro, ainsi que les dates des trimestres correspondants (cf le fichier en question pour connaître l'ordre exact). L'échantillon s'étend de 1970-I à 2009-IV.

1a Récupérer les séries et créer les vecteurs colonnes `t`, `Q`, `P` et `r` représentant respectivement les dates, le PIB, le niveau des prix et le taux d'intérêt de la zone euro `[load]`.

```
donnees = load('./data/ZEdata.txt');
t = donnees(:,1);
Q = donnees(:,2);
P = donnees(:,3);
r = donnees(:,4);
```

1b La variable de prix est exprimée en niveau, sous forme d'indice trimestriel. Calculer `pie`, le taux d'inflation correspondant (sans le multiplier par 100).

```
pie = diff(log(P));
```

1c Le taux d'intérêt est exprimé en % et en taux annuel. Transformer la série pour obtenir un taux trimestriel en niveau qu'on appelera `r`.

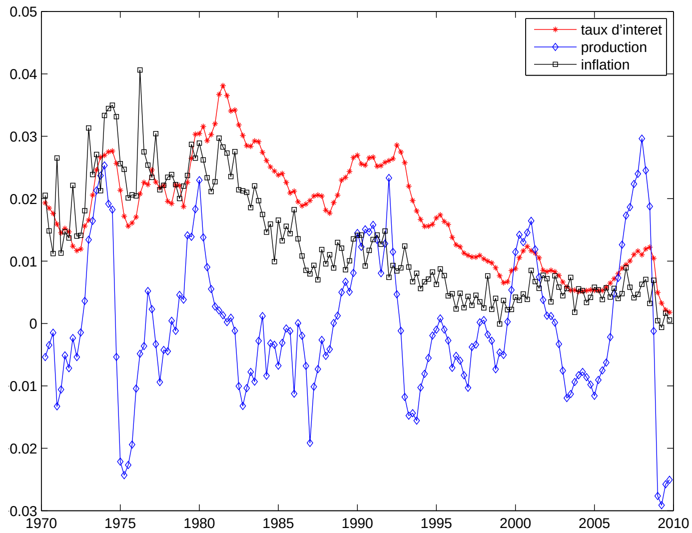
```
r = (1+r(2:end)/100).^(1/4)-1;
```

1d Identifier la composante cyclique du PIB en logarithme qu'on notera `y` `[hptrend]`.

```
lY = log(Q(2:end));
[crap,y] = hptrend(lY,1600);
```

1e Représenter sur un même graphique le taux d'intérêt `r`, la production `y` et l'inflation `pie`. Commenter.

```
t = t(2:end);
plot(t,r,'-r*',t,y,'-bd',t,pie,'-ks','MarkerSize',4);
legend('taux d''interet','production','inflation');
print('./pdf/fig_rypie.pdf','-dpdf')
```



1f Calculer la composante cyclique de l'inflation qu'on notera `infl` en considérant une tendance quadratique déterministe.

On va extraire une tendance déterministe linéaire des séries d'inflation et du taux d'intérêt. On définit les regresseurs : une tendance t et une constante 1, soit la matrice $X_t = [t^2 \; t \; 1]$. On cherche le vecteur β dans

$$Y_t = X_t\beta + \epsilon_t$$

avec $\epsilon_t \sim$ i.i.d. $N(0, \sigma^2)$. On sait que l'estimateur de β s'écrit

$$\hat{\beta} = (X'X)^{-1}X'Y$$

```
T = length(t);   % taille de l'échantillon
trend = (1:T)';
XX = [trend.^2 trend ones(T,1)];
YY = pie;
betp = (XX'*XX)\(XX'*YY) ;
disp('betp');
disp(betp);

betp
    0.0000
   -0.0002
    0.0273

betr
   -0.0000
    0.0002
    0.0189
```

1g Appliquer la même procédure qu'en 1f au taux d'intérêt. On notera `rate` la composante cyclique ainsi identifiée.

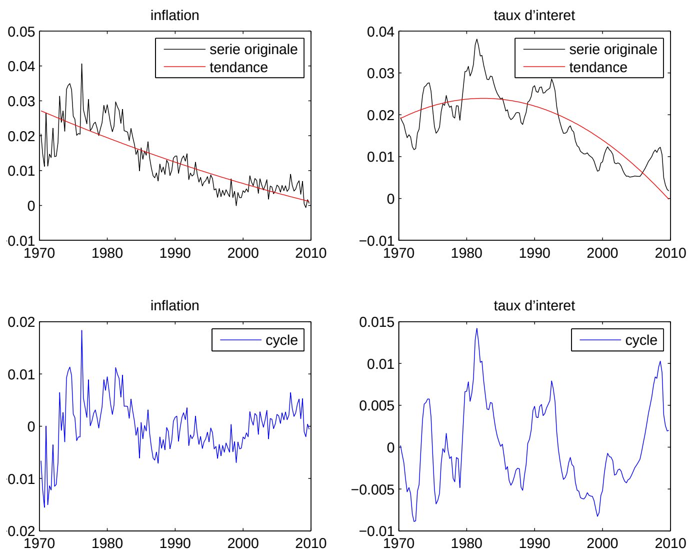
```
YY = r ;
betr = (XX'*XX)\(XX'*YY);
disp('betr');
disp(betr);
```

On retrouve bien des tendances négatives pour l'inflation et le taux d'intérêt. On récupére les composantes cycliques

```
infl = pie - XX*betp ;
rate = r - XX*betr ;
```

1h Représenter pour l'inflation et le taux d'intérêt, la série originale et la tendance (sur un même graphique), et la composante cyclique `[subplot,plot]`.

```
figure;
subplot(2,2,1); plot(t,pie,'k',t,pie-infl,'r');
legend('serie originale','tendance'); title('inflation');
subplot(2,2,2); plot(t,r,'k',t,r-rate,'r');
legend('serie originale','tendance'); title('taux d''interet');
subplot(2,2,3); plot(t,infl);
legend('cycle'); title('inflation');
subplot(2,2,4); plot(t,rate);
legend('cycle'); title('taux d''interet');
print('./pdf/fig_inflrate.pdf','-dpdf')
```



Estimation d'un VAR

Dans cette partie, on cherche à estimer un VAR(4) pour le PIB, l'inflation et le taux d'intérêt de la zone euro. Ce var s'écrit

$$B(L)\,Y_t = e_t$$

où $B(L)$ est un polynôme de degré 4, L est l'opérateur retard, et e_t un vecteur d'innovation, tel que $e_t \sim \mathcal{N}(0, \Sigma)$. Y_t est le vecteur des observables $Y_t = (y_t, \pi_t, r_t)$. (Attention, l'ordre compte !)

2a Créer la variable `Y` qui empile en ligne les observations, et en colonne les variables.

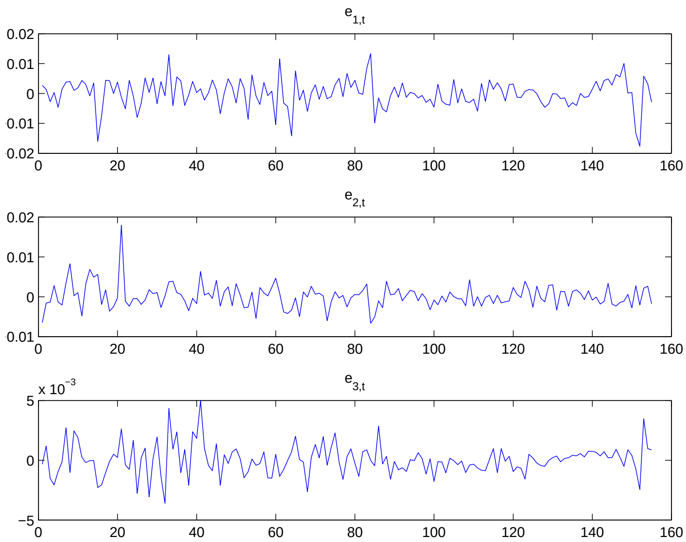
```
Y = [y infl rate];
```

2b Estimer le VAR(4) en utilisant la commande de Sims `rfvar3`. Observer la valeur des coefficients. Commenter. En particulier, y a-t-il des coefficients nuls ?

```
p = 4 ;
var = rfvar3(Y,p,[],[],[],[]);
var.By

ans(:,:,1) =
    1.0602   -0.0316    0.7414
    0.1335    0.2643    0.7167
    0.0901    0.0259    1.2371

ans(:,:,2) =
   -0.1930    0.0410   -0.9205
   -0.0977    0.1951   -0.5142
   -0.0361   -0.0011   -0.4546

ans(:,:,3) =
   -0.0125   -0.2262    0.3200
   -0.0830   -0.0424    0.1033
   -0.0204    0.0091    0.1132

ans(:,:,4) =
   -0.0817    0.1010   -0.2071
    0.0760    0.3649   -0.2863
   -0.0055   -0.0055    0.0231
```

2c Représenter graphiquement les résidus de chaque équation. Paraissent-ils stationnaires ?

```
figure;
for ii = 1:3
    subplot(3,1,ii); plot(var.u(:,ii));
    title(['e_{' num2str(ii) ',t}']);
end;
print('./pdf/fig_e.pdf','-dpdf')
```



2d Calculer la matrice de variance-covariance sous forme matricielle, et comparer avec la fonction `cov`.

Calcul de la matrice de variance-covariance Σ_e. On l'obtient à partir des residus du VAR, c.-à-d. des innovations des 3 équations en appliquant le calcul suivant :

$$\Sigma_e = \frac{1}{T} e_t' e_t$$

```
Sig_e = (var.u'*var.u)/(T-p-1) ;
disp('Sig_e =');
fprintf('\n');
disp(Sig_e);
Sig_e =
   1.0e-04 *
    0.2268    0.0022    0.0199
    0.0022    0.0854    0.0078
    0.0199    0.0078    0.0167
```

à comparer avec la fonction `cov`

```
disp(cov(var.u));
1.0e-04 *
 0.2268    0.0022    0.0199
 0.0022    0.0853    0.0078
 0.0199    0.0078    0.0167
```

Fonction de réponse

Afin d'identifier les impulsions implicites dans le modèle, nous allons utiliser la procédure d'identification par le facteur de Cholesky.

3a Calculer le facteur de Cholesky `[chol]`.

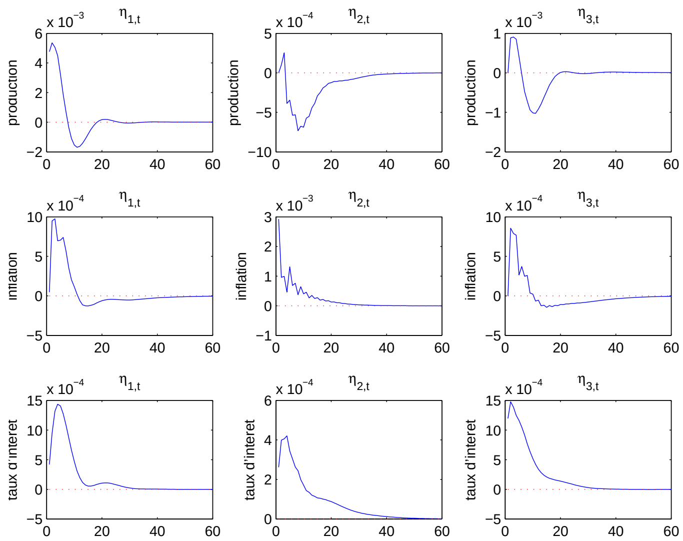
```
S = chol(Sig_e);
```

3b Calculer les IRF. On prendra pour horizon 60 periodes [impulsdtrf].

```
nstep = 60 ;
IRF = impulsdtrf(var.By,S,nstep);
```

3c Représenter les IRFs pour chaque variable et pour chaque choc [subplot]. Commenter.

```
figure;
lbl = {'production','inflation','taux d''interet'};
n = 1 ;                          % numéro du graphe
for ii = 1:3                     % nombre de variables
    for jj = 1:3                 % nombre de chocs
        subplot(3,3,n);
        tmp = IRF(ii,jj,:);
        plot((1:nstep),tmp(:),(1:nstep),zeros(1,nstep),':r');
        title(['\eta_{' int2str(jj) ',t}']);
        ylabel(char(lbl(ii)));
        n = n + 1;
    end
end
print('./pdf/fig_irf.pdf','-dpdf')
```



Intervales de confiance

La série observée n'est qu'une réalisation d'un processus stochastique, issue par hypothèse d'un processus générateur des données (le modèle VAR) et d'une chronique de chocs (les e_t). En réutilisant ces chocs et en les mélangeant, on devrait obtenir les mêmes IRFs, à l'incertitude sur les e_t près. Nous allons simuler des séries à partir des p premières valeurs de Y_t et d'un mélange de séries $\{e_t^{(i)}\}_{i=1}^N$.

4a Récupérer les résidus dans la variable `e` et créez une procédure permettant de mélanger ces résidus pour engendre un nouveau vecteur `e_sim [rand]`.

```
residus = var.u ;
e_tmp = residus(randi(T-p,T-p,1),:);
```

4b Simuler une série `Y_sim` à partir des $p = 4$ premières valeurs de `Y` et de la chronique de chocs `e_sim`.

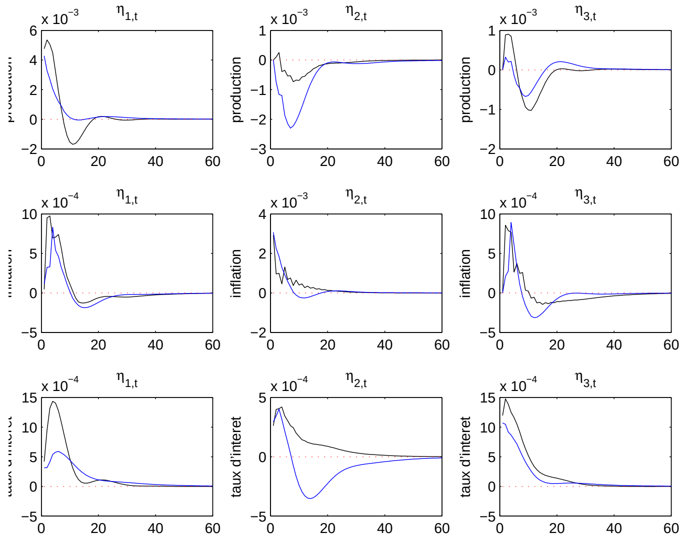
```
Y_sim = zeros(size(Y));
Y_sim(1:p,:) = Y(1:p,:);
for ii = p+1:T
    for jj = 1:p
        Y_sim(ii,:) = Y_sim(ii,:) + Y_sim(ii-1,:)*var.By(:,:,jj)';
    end
    Y_sim(ii,:) = Y_sim(ii,:) + e_tmp(ii-p,:);
end
```

4c Estimer à nouveau un VAR(4) et recalculer les IRF. Comparer pour une variable et un choc les deux IRF (l'originale et la simulée).

```
var_tmp = rfvar3(Y_sim,p,[],[],[],[]);
Sig_e_tmp = (var_tmp.u'*var_tmp.u)/(T-p-1) ;
IRF_tmp = impulsdtrf(var_tmp.By,chol(Sig_e_tmp),nstep);
```

On peut voir ce que ça donne avec deux IRFs.

```
n = 1 ;                         % numéro du graphe
for ii = 1:3                    % nombre de variables
    for jj = 1:3                % nombre de chocs
        subplot(3,3,n);
        tmp1 = IRF(ii,jj,:);
        tmp2 = IRF_tmp(ii,jj,:);
        plot((1:nstep),tmp1(:),'k',...
             (1:nstep),tmp2(:),'b',...
             (1:nstep),zeros(1,nstep),':r');
        title(['\eta_{' int2str(jj) ',t}']);
        ylabel(char(lbl(ii)));
        n = n + 1;
    end
end
print('./pdf/fig_2irf.pdf','-dpdf')
```



A priori, quelque chose de proche, mais pas trop. Pour créer un intervale de confiance, on va répéter le processus et engendrer beaucoup d'IRF. On calculera ensuite pour chaque horizon les 5% supérieur et les 5% inf.

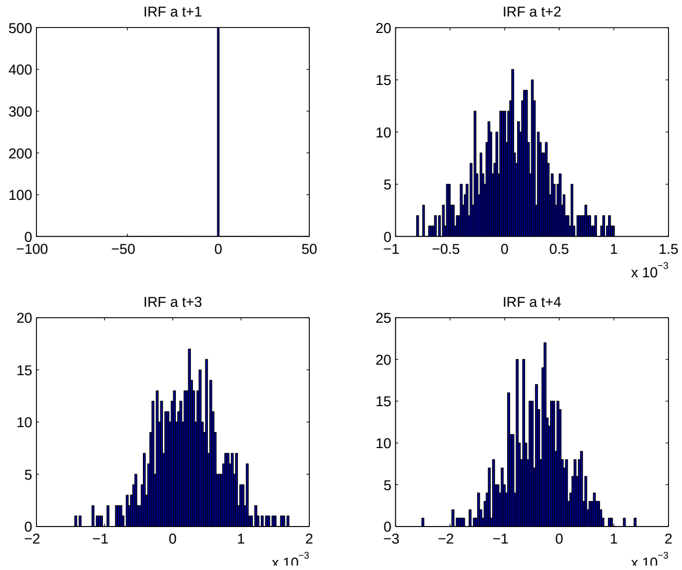
4d Proposer une structure de boucle pour simuler `nsim` fois cette procédure. Qu'obtient-on ? Proposer une méthode de calcul de la valeur des centiles à 5% et à 95%.

```
% initialisation
h = waitbar(0,'Please wait...');
nsim = 500 ;                        % d'après Hamilton, plutôt 10000
BIRF = zeros(3,3,nstep,nsim);

for nn = 1:nsim
    waitbar(nn/nsim,h)
    e_tmp = var.u(randi(T-p,T-p,1),:);
    Y_sim = zeros(size(Y));
    Y_sim(1:p,:) = Y(1:p,:);
    for ii = p+1:T
        for jj = 1:p
            Y_sim(ii,:) = Y_sim(ii,:) + Y_sim(ii-jj,:)*var.By(:,:,jj)';
        end
        Y_sim(ii,:) = Y_sim(ii,:) + e_tmp(ii-p,:);
    end
    var_tmp = rfvar3(Y_sim,p,[],[],[],[]);
    Sig_e_tmp = (var_tmp.u'*var_tmp.u)/(T-p-1) ;
    BIRF(:,:,:,nn) = impulsdtrf(var_tmp.By,chol(Sig_e_tmp),nstep);
end
close(h);
```

Pour chaque horizon, on a une distribution d'IRFs, par exemple pour les 4 premiers horizons de l'IRF du 1er choc sur la 1re variable :

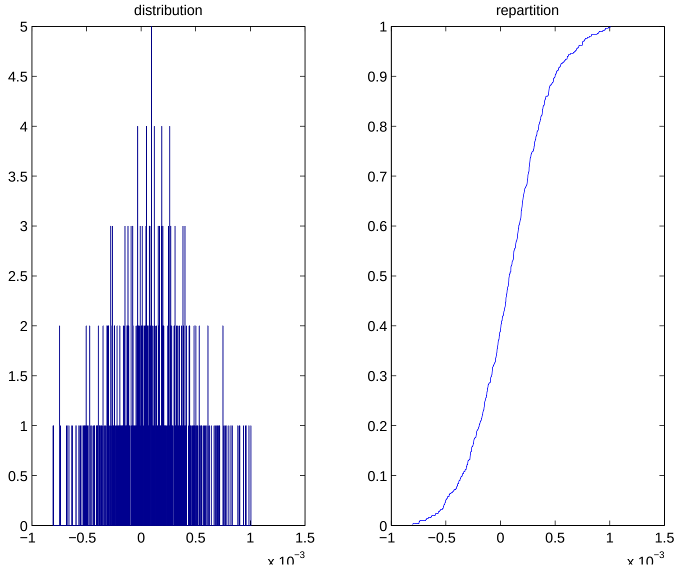
```
for ii = 1:4
```

```
    tmp = BIRF(1,2,ii,:);
    subplot(2,2,ii); hist(tmp(:),100);
    title(['IRF a t+' int2str(ii)]);
end
print('./pdf/fig_distribirf.pdf','-dpdf')
```



Pour calculer les bornes à 5%, il faut la fonction de répartition des valeurs à chaque horizon. On va utiliser pour cela la fonction `hist`. Par exemple, pour l'effet du 1er choc sur la 1re variable à la 1re periode :

```
tmp = BIRF(1,2,2,:);          % attention à la dimension de tmp
tmp = tmp(:);                 % tmp est à présent un vecteur colonne
[N,val] = hist(tmp,1000);
F = cumsum(N)/nsim;
figure
subplot(1,2,1); hist(tmp,1000); title('distribution');
subplot(1,2,2); plot(val,F); title('repartition');
print('./pdf/fig_hist.pdf','-dpdf')
```



Avec cette fonction de répartition F, on va pouvoir récupérer la valeur de l'IRF pour lequel on a 5% en dessous, ou 95% en dessous.

```
irf_min = min(val(F>=.05));
irf_max = min(val(F>=.95));
```

et on obtient

```
disp('irf_min =');
fprintf('\n');
disp(irf_min);
fprintf('\n');
disp('irf_max =');
fprintf('\n');
disp(irf_max);
fprintf('\n');
irf_min =
  -4.4078e-04

irf_max =
   6.4457e-04
```

On fait ça pour toutes les IRFs et on stocke les valeurs de IRF_min et IRF_max dans des matrices.

```
IRF_min = zeros(3,3,nstep);
IRF_max = zeros(3,3,nstep);

for ii = 1:3
    for jj = 1:3
        for ss = 1:nstep
            tmp = BIRF(ii,jj,ss,:);
            tmp = tmp(:);
            [N,val] = hist(tmp,1000);
```

```
            F = cumsum(N)/nsim;
            IRF_min(ii,jj,ss) = min(val(F>=.05));
            IRF_max(ii,jj,ss) = min(val(F>=.95));
        end
    end
end
```

On peut retracer alors les IRF avec les intervales de confiance.

```
figure;
lbl = {'production','inflation','taux d''interet'};
n = 1 ;                         % numéro du graphe
for ii = 1:3                    % nombre de variables
    for jj = 1:3                % nombre de chocs
        subplot(3,3,n);
        tmp = IRF(ii,jj,:);
        tmp_min = IRF_min(ii,jj,:);
        tmp_max = IRF_max(ii,jj,:);
        hor = (1:nstep);
        plot(hor,tmp(:),...
             hor,tmp_min(:),'r',...
             hor,tmp_max(:),'r',...
             hor,zeros(1,nstep),':r');
        title(['\eta_{' int2str(jj) ',t}']);
        ylabel(char(lbl(ii)));
        n = n + 1;
    end
end
print('./pdf/fig_conf.pdf','-dpdf')
```